This is the author's final version of the contribution published as:

F. Cena; S. Likavec; F. Osborne. Propagating user interests in ontology-based user model, in: LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER, 2011, 9783642239533, pp: 299-311.

The publisher's version is available at:
http://link.springer.com/content/pdf/10.1007/978-3-642-23954-0_28

When citing, please refer to the published version.

Link to this full text:
http://hdl.handle.net/2318/130769

# Propagating user interests in ontology-based user model

Federica Cena, Silvia Likavec, Francesco Osborne

Dipartimento di Informatica, Università di Torino, Italy
{cena,likavec,osborne}@di.unito.it *

**Abstract.** In this paper we address the problem of propagating user interests in ontology-based user models. Our ontology-based user model (OBUM) is devised as an overlay over the domain ontology. Using ontologies as the basis of the user profile allows the initial user behavior to be matched with existing concepts in the domain ontology. Such ontological approach to user profiling has been proven successful in addressing the cold-start problem in recommender systems, since it allows for propagation from a small number of initial concepts to other related domain concepts by exploiting the ontological structure of the domain. The main contribution of the paper is the novel algorithm for propagation of user interests which takes into account i) the ontological structure of the domain and, in particular, the level at which each domain item is found in the ontology; ii) the type of feedback provided by the user, and iii) the amount of past feedback provided for a certain domain object.

**Keywords:** user model, ontology, propagation of interests

## 1 Introduction

In different areas of the Web, personalization and adaptation are crucial concepts nowadays, since they help users find what they really want and need. From e-commerce to e-learning, from tourism and cultural heritage to digital libraries, users benefit from tailoring the content and visualization techniques to their own needs. A simple way to capture the user preferences and/or interests and account for differences in needs of individual users is provided by user modeling. In user-adaptive and recommender systems [7, 1], a User Model stores the available information about a user by maintaining user properties such as interests, preferences, knowledge, goals and other facts considered relevant for the application. The information in the user model is then used by adaptive systems, applying some reasoning strategies, to provide personalization services to each user (e.g. adapting the interface or the contents order, or recommending certain items to users).

User models can be constructed in different ways, e.g. by exploiting the information provided by users upon registration, clustering users in stereotypes, obtaining the information from other applications, deriving information from users' behavior etc. The latter case is particularly relevant, since users' interaction with the system can be implicitly monitored and recorded by the system, providing a rich source for further analysis with minimal user intervention.

* This work has been supported by PIEMONTE Project - People Interaction with Enhanced Multimodal Objects for a New Territory Experience.

There are several approaches to representing user models, from simple property-value pairs, to more complex probabilistic approaches, such as Bayesian networks. Very often the user model is conceived as an overlay over the domain model, where the user's current state (such as interest or knowledge) with respect to domain concepts is recorded [3]. For each domain model concept, an individual overlay model stores a value which is an estimation of the user's attitude to this concept. For example, an overlay model of user interests can be represented as a set of pairs $\langle concept, value \rangle$, with one pair for each domain concept.

Ontologies as explicit specifications of domain concepts and relationships between them are emerging as powerful formalisms for knowledge representation with associated reasoning mechanisms. In the last years, there is a growing tendency to use ontologies to represent the domain models. In this context, one of the promising approaches to represent a user model is by conceiving it as an *overlay over the domain ontology*.

In this paper we address the problem of propagating user interests in ontology-based user models. Our ontology-based user model (OBUM) is devised as an overlay over the domain ontology. Using ontologies as the basis of the user profile allows the initial user behavior to be matched with existing concepts in the domain ontology. Such ontological approach to user profiling has been proven successful in addressing the cold-start problem in recommender systems, since it allows for propagation from a small number of initial concepts to other related domain concepts by exploiting the ontological structure of the domain [9].

The main contribution of the paper is the novel algorithm for propagation of user interests which takes into account:

- the ontological structure of the domain and, in particular, the level of the object receiving the user feedback in the ontology;
- the type of feedback provided by the user;
- the amount of past feedback provided for a certain domain object.

In addition, our approach allows for bi-directional propagation of user interests in the ontology, i.e. both bottom-up and top-down. This approach contributes to resolution of the cold-start problem, thus improving the adaptation of the system at the beginning. It also alleviates the sparsity problem (i.e., the much bigger number of items than the number of items rated by users) which often plagues recommender systems.

The rest of this paper is organized as follows. We begin by presenting our specific algorithm for propagating user interests on an ontology in Sect. 2. Then, we describe how we apply our approach on an existing application in Sect. 3, followed by the results of a preliminary evaluation in Sect. 4. In Sect. 5 we present some related work and background. Finally, we conclude and give some directions for future work in Sect. 6.

## 2 Our Approach

In this paper, we propose an approach to propagating user interests in a domain ontology, starting from the user behavior in the system. Our approach is based on the following requirements:

- the domain model has to be represented using an *ontology*, where domain items are modeled as instances of the concepts in the ontology;[1]
- the user model has to be represented as an overlay over the domain ontology;
- user interaction with the system has to be recorded in order to infer user interests.

We chose to use ontologies to represent the objects of the domain for several reasons. Ontologies guarantee exact semantics for each statement, and avoid semantic ambiguities. Since they are expressed with standard formats and technologies, they allow for extensibility and re-usability. But most importantly, their structure allows for powerful and rigorous reasoning over data (inheritance, subsumption, classification etc.).

Regarding the user model, we decided to employ an *ontology-based user model*, where user interests are recorded for the classes of the domain ontology. Each ontological user profile is an instance of the reference ontology where every domain object in the ontology has an interest value associated to it. This means that each node $N$ in the domain ontology can be seen as a pair $\langle N, \mathcal{I}(N)\rangle$, where $\mathcal{I}(N)$ is the interest value associated to $N$ denoting the user interest in the object represented with the node $N$.

As for the last point, we chose to infer user interests indirectly by observing user behavior with the system. When interacting with the system, users provide valuable feedback about their interests that the system records implicitly and can use to incrementally create (and update) the user model by modifying the interest values for certain domain objects. According to Kobsa [7], possible user actions are: selecting, tagging, rating, commenting or bookmarking an item. Each of these actions is assigned a certain weight $\mathsf{f}$, according to its strength as a signal of user interest. For example, clicking on a certain domain object denotes less interest than bookmarking it. All these actions are being registered in the log files in order to permit for their later retrieval and further analysis. Our idea is to use this feedback to:

- infer user interest in an object receiving the feedback;
- calculate the interests in other related domain objects, such as ancestors or descendants.

Each time a user provides a feedback, the following steps take place. First, we calculate the level of interest in the node that received the feedback (*sensed interest*). The sensed interest is added to the initial node interest and used for the subsequent propagation phase. Then, starting from this value, we calculate the *propagated interest* for the nearest nodes. During the propagation, the algorithm traverses vertically the ontology graph, and for each node it meets, the original interest is incremented by the *propagated interest*, which depends on the sensed interest, the distance from the initial node and the amount of the past feedback received.

The user interest $\mathcal{I}$ in a certain item is calculated as follows:

- for the node which receives direct user feedback:

$$\mathcal{I}(N) = \mathcal{I}_{\mathsf{o}}(N) + \mathcal{I}_{\mathsf{s}}(N) \quad (1)$$

- for the node which receives an interest value propagated vertically:

$$\mathcal{I}(M) = \mathcal{I}_{\mathsf{o}}(M) + \mathcal{I}_{\mathsf{p}}(M) \quad (2)$$

where

- $\mathcal{I}_{\mathsf{o}}$ (**old interest**) is the old value for the user interest (initially equal to zero);
- $\mathcal{I}_{\mathsf{s}}$ (**sensed interest**) is the value obtained from the direct feedback of the user;

[1] For the time being, our method is designed for a static domain, where dynamic modification of the domain is not managed.

– $\mathcal{I}_{\mathsf{P}}$ (**propagated interest**) is the value obtained by vertical propagation.

We explain below how to calculate sensed interest and propagated interest.

### 2.1 Inferring user interest (sensed interest)

*Sensed interest* is the value that shows how much of a direct feedback from the user the given node "senses". Thus, we introduce the concept of a *sensitivity* of a node. The sensitivity of a certain node depends on its position in the ontology: if the user provides feedback about the node lower down in the ontology, the effect is stronger than when the feedback is received for the node higher up, requiring a lower amount of feedback as a signal of strong user interest for a lower node. In fact, since the lowest nodes in the ontology represent specific concepts, they signal more precise interest than interest expressed in more general concepts, represented by upper classes in the ontology. For example, declaring interest in `Sparkly_White_Wine` gives more precise information with respect to declaring interest in generic item `Wine`).

In order to calculate the interest sensed by a given node N, we use the following formula (adaptation of Stevens' power law [18] used to relate the intensity of the physical stimulus and its sensed intensity):

$$\mathcal{I}_{\mathsf{S}}(N) = \frac{l(N)+1}{max+1}\mathsf{f}(N)^b \tag{3}$$

where $l(N)$ is the level of the node that receives the feedback, $max$ is the level of the deepest node in the ontology, $\mathsf{f}(N)$ is the feedback obtained from the user for the node $N$, $b \in \mathbb{R}$ is a constant $(0 < b < 1)$ which controls how strongly the node senses each different type of direct feedback. Thus, it is possible to account for the case where one type of feedback has a stronger impact than the other, but also to keep $b$ constant so as to perceive all user actions equally.

For example, if a leaf node $N$ ($l(N) = max$) receives the first feedback $\mathsf{f} = 5$ its sensed interest is $\mathcal{I}_{\mathsf{S}}(N) = 5^b$. Since $\mathcal{I}_{\mathsf{O}}(N) = 0$, the cumulative interest $\mathcal{I}(N) = 5^b$. When the same node receives the second feedback $\mathsf{f} = 10$, $\mathcal{I}_{\mathsf{S}}(N) = 10^b$ and $\mathcal{I}(N) = 5^b + 10^b$.

### 2.2 Propagating user interest (propagated interest)

The main idea is that the effect of a feedback for a given node (i.e. the interest in a certain domain object) can be propagated *vertically* in the ontology, upward to the ancestors as well as downward to the descendants of a given node[2]. For example, if a person is interested in red wine `Barbera_d'Asti`, it is safe to assume that the same person might be interested in a specific kind of this wine, such as `Vietti_Barbera_d'Asti_Tre_Vigne_2008`, but also in `Red_Wine` in general. Of course, the interest for an object that can arise from this assumption is less strong than the original and will depend on the conceptual distance between the two nodes. Therefore we can assume that the interest we can propagate vertically is inversely correlated to the distance from the node that

[2] This idea is based on the similarity derived from IS-A relations. Concepts connected by IS-A relations are somehow similar, according to taxonomy based approach [13], since descendant concepts are subclasses of a class and the subclasses inherit attributes of the upper classes.

received the feedback. In order to correctly propagate the interests, the contributions of the various nodes to the propagation must be balanced. In particular, the node contribution to the interest propagation process should be decreasing with the amount of feedback already spread by that node. This is needed to prevent the non-proportional propagation of interests when the user concentrates solely on one or a few items. In that case, this input is redundant and does not add any information. On the contrary, we want to reward those cases where an ancestor node receives feedback from a good part of its sub-nodes, showing a consistent interest for that class.

The *propagated interest* is the value of “indirect” interest that a node can receive as a result of vertical propagation in the ontology. It is calculated modulating the sensed interest $\mathcal{I}_{\mathsf{S}}(N)$ of the node $N$ that receives the feedback by the exponential factor which describes the attenuation with each step up or down (simulating attenuation in physics), and a weight inversely correlated with the amount of feedback already received by the node as follows:

$$\mathcal{I}_{\mathsf{P}}(M) = \frac{e^{-kd(N,M)}}{1 + \log(1 + n(M))}\mathcal{I}_{\mathsf{S}}(N) \tag{4}$$

where $d(N, M)$ is the distance between the node $N$ receiving the feedback and the node $M$ receiving the propagated interest, $n(M)$ is the number of actions performed in the past on the node M and $k \in \mathbb{R}$ is a constant. Varying the attenuation coefficient $k$, it is possible to control how much interest is propagated depending on the type of feedback received: for instance, favoring the feedback resulting from bookmarking rather than the feedback from simply visiting the page.

## 3 Use Case

We exploited our approach in gastronomic domain, using the *WantEat* application [10] as a use case. *WantEat* is a part of an ongoing project which aims at providing a ”Social Web of Entities”, where a network of people and intelligent objects is built to enable their mutual interaction. People can interact in a natural way with these objects, accessing information and stories about them. They can navigate social networks of objects and people, annotate, rate, bookmark and comment the objects. The behavior of the system is adaptive, since it personalizes the interaction according to the preferences of individual users. In particular, the order of the objects presented to the users takes into account the user model.

The domain of the project is gastronomy: enhanced objects can be products such as cheeses and wines, as well as places of origin, shops, restaurants, market stalls etc. Such a domain is represented using an upper ontology which imports additional sub-ontologies describing particular areas of the domain (products, recipes, actors, places). The domain objects are modeled as the instances of the classes of the ontology.

Following the requirements of our approach described in Sect. 2, the user model is represented as an overlay over such an ontology, associating, for each user, an interest value for every domain class in the ontology. Such interests are derived by the system from the feedback the user provides in his interaction with a given object, which can be a certain category (e.g. wine, cheese, etc.) or a specific item (e.g. a specific wine or cheese). There are five possible typologies of user feedback in the system: (i) clicking on an object, (ii) tagging, (iii) commenting, (iv) voting on a scale from 1 to 5, or (v) putting an item into favorites. Each type of feedback has different impact on the user model, based on the strength of interest indication. The lowest interest indicator is obtained from clicking on a certain domain item, followed by commenting, tagging, voting and putting an item into favorites (this is the indicator of the highest interest). All these actions are registered in the log files and analyzed. According to the strength of interest indication, each of these actions is assigned a certain weight $\mathsf{f}$, following the approach developed in [4]. For example, clicking on a certain domain object can have $\mathsf{f} = 3$, whereas tagging can have $\mathsf{f} = 8$ (see Table 1).

| **Action** | **Weight** |
|---|---|
| Bookmarking an object | 9 |
| Tagging an object | 7 |
| Commenting on an object | 5 |
| Rating an object | 1*vote |

**Table 1.** Weights associated to user actions

Hence, the users indicate their interests directly for the domain objects. We want to use these values to calculate users' interests in such objects, but also the interests in some other related domain objects.

After having collected the users' feedback, we applied our approach for inferring interests and propagating them, taking into account (as explained in Sect. 2):

1. the type of feedback provided by the user (see Table 1);
2. the amount of feedback (the number of actions the user performed while interacting with the system);
3. the level in the ontology of the object that received the feedback.

As an example, let us consider the following scenario. Tom uses the application to find a good restaurant nearby and he discovers a restaurant famous for its cold cuts. Since he particularly likes ham, he bookmarks `Prosciutto_Crudo` (raw ham). With this information, the system is able to a) infer a value of interest for this class, and b) infer the level of interest in the related classes.

The bookmarking process assigns a feedback of 9 to the class `Prosciutto_Crudo` (according to Table 1). `Prosciutto_Crudo` is a leaf at the maximum level of the ontology ($l(PC) = max$, see Fig. 1) and we assume $b = 0.5$. Thus, $\mathcal{I}(PC) = \mathcal{I}_{\mathsf{S}}(PC) = 9^{0.5} = 3$. Starting from this sensed interest, we propagate this value according to (4). Since the class is a leaf, the propagation is directed only to the ancestor classes.

The first ancestor class we meet is `Salumi_Interi_Crudi` (whole raw cold cuts) (see Fig. 1). We assume that the distance between the nodes is equal to the number of edges between them (1 in this case) and $k = 0.8$. Besides, the class `Salumi_Interi_Crudi` did not receive any feedback in the past. The system will derive the interest of the node $\mathcal{I}(SIC) = \mathcal{I}_{\mathsf{p}}(SIC) = \mathcal{I}_{\mathsf{S}}(PC)e^{-0.8} = 1.35$. The second ancestor class we meet is `Salumi_Interi`. Let us assume that this class has already received 4 feedback in the past. Hence, it will receive $\mathcal{I}_{\mathsf{p}} = 3e^{-1.6}/1.7 = 0.36$. The propagation phase stops when we get to the root.

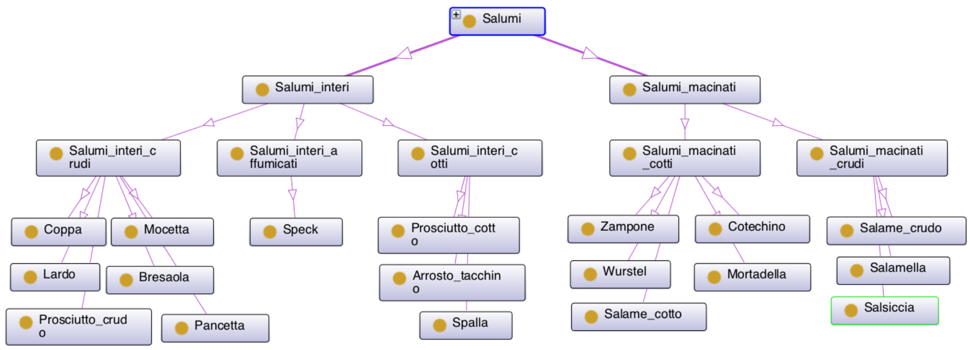


**Fig. 1.** The portion of domain ontology representing cold cuts.

The same process occurs for all other feedback. Of course, using other values for $k$ and $b$ we can obtain different propagation behavior. We can see that, as Tom simply expresses his interest for `Prosciutto_Crudo`, the system starts to infer his interest in all the correlated classes. After a little feedback the application will be able to suggest to Tom other objects he probably likes: for example, it will know that he prefers `Salumi_Interi_Crudi` to `Salumi_Interi_Cotti` (whole cooked cold cuts). The propagation process works also the other way around, hence the application will be ready to suggest similar objects like `Speck`, using the feedback Tom may have given on a ancestor class like `Salumi_Interi_Affumicati` (whole smoked cold cuts).

## 4 Evaluation of the Approach

We used the application described in the previous section to test our approach for propagating interest values in the ontology. In particular, the evaluation was performed on the portion of the ontology described in the above scenario (see Fig. 1). One of the most important personalization operations performed by the application is the ordering of objects returned by a search or displayed during the navigation. Producing a list that mirrors the user's preferences is a challenging task which provided a valuable testing environment for our algorithm.

Note that in our approach, we consider both the direct feedback provided for a certain object and the propagated interest. However, in the evaluation we decided to only focus on the contribution provided by vertical propagation. This may be thought of as a borderline case, in which we have no feedback on the objects to order, thus we have to rely solely on the propagation technique presented above.

**Hypothesis.** We assumed that our algorithm can be used to predict user's interests in certain objects of a given domain, starting from various classes in the ontology and propagating the interest to the related concepts (ancestors and descendants of a given class). We wanted to compare the lists of domain items generated by our algorithm with the ordered lists provided by the users themselves.

**Experimental Design.** We designed a questionnaire to collect users' preferences. It was divided into two parts, in order to test both upward and downward propagation.

In the first part, the users were asked to rate 8 non-leaf classes of our ontology (e.g. `Salumi_Interi_Cotti`) on a scale from 1 to 10. In the second part, they had to indicate 4 leaves (e.g. `Prosciutto_Crudo`, `Speck` etc.) they "like very much" (simulating the bookmarking action) and four leaves they "like enough" (simulating the tagging action) among the total of 15 leaf classes.

**Subjects.** The sample included 92 subjects, 19-45 years old, recruited according to an availability sampling strategy[3].

**Measures and Material.** Users were asked to connect to the web site with written instructions and compile the anonymous questionnaire. Users' ratings were registered in a database. The evaluation followed two phases:

– We tested the *downward* propagation by generating a list of leaves based on the feedback provided for the higher classes and comparing it with the list provided by the user.
– We tested the *upward* propagation by generating a list of classes based on the feedback provided for the leaves and comparing it with the list provided by the user.

To compare the two lists we used the Pearson's correlation coefficient $r$ between ranks, that measures the degree of association between two ordered lists of items[4].

**Results.** We have collected the total of 1,472 ratings and we calculated Pearson's correlation coefficient for 92 pairs of lists. Note that $-1 \leq r \leq 1$, where $r = 1$ corresponds to perfect association (the two lists follow the same order) and $r = -1$ corresponds to perfect inverse association.

*a) Downward propagation.* Since the leaves that could be voted were eight, whereas the options to signal the degree of user interest were only two, we have several tied ranks and the degree of association $r$ in the downward propagation test is restricted to 5 values. These values correspond one to one to the number of errors in the generated lists: $r = 1$ corresponds to no errors, $r = 0.5$ to 1 error, $r = 0$ to 2 errors, and so on. Figure 2 shows the distribution of cases for various values of $r$. Notice that in 73% of the cases we were able to generate the lists with no more than 1 error. In almost one fifth of the cases we were able to generate the list equal to the user list. Only 25% of the cases had no benefit from the technique. The application of the downward propagation has almost never negative effects: only two cases showed a medium inverse association and not one a strong inverse association.

*b) Upward propagation.* The ordered lists used for testing the upward propagation were constructed by choosing 4 bottom classes the users "liked very much" and 4 bottom classes the users "liked enough". We assigned numerical values on a scale from 1 to 10 to "like very much" (9) and "like enough" (7) to simulate bookmarking and tagging. Then we generated personalized lists using these 8 feedback. Figure 3 shows the distribution of cases for various association coefficients r, which in this case cover 20 values. We can see that in 88% of the cases our algorithm was able to generate a list with a positive association with the one provided by the user. In 61% of the cases

[3] Even though random sampling is the best way of having a representative sample, these strategies require a great deal of time and money. Therefore much research in social science is based on samples obtained through non-random selection, such as the availability sampling, i.e. a sampling of convenience, based on subjects available to the researcher, often used when the population source is not completely defined.

[4] In this case it is not correct to talk about linear correlation since there are no independent and dependent variables related by a linear relation.

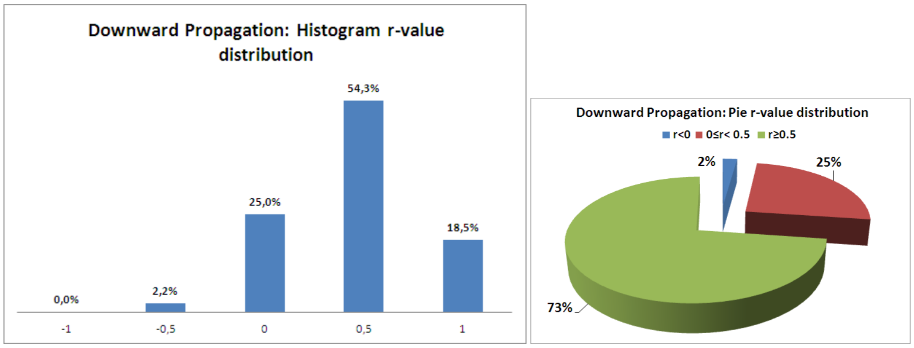


**Fig. 2.** Upward propagation: the distribution of cases for various values of r

$r \geq 0.5$. More impressive, in 27% of the cases we obtained $r \geq 0.9$, value which is usually considered "very good". Only in 12% of the cases $r < 0$.

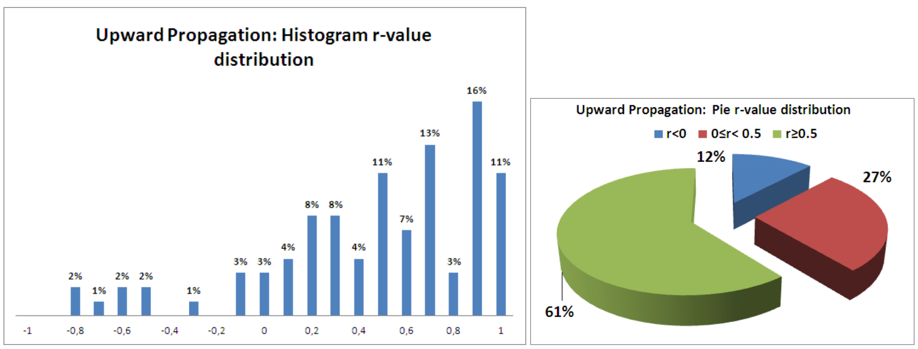


**Fig. 3.** Upward propagation: the distribution of cases for various values of r.

**Discussion.** The preliminary tests described above aimed at evaluating the possibilities of our algorithm, in particular a propagation technique that works both upwards and downwards. Both propagation modalities gave positive results. For a great number of the cases we were able to generate a perfect list or a list with a high degree of association. The cases that could present the user with a misleading list are practically zero for the downward propagation and very low for the upward one.

## 5 Related Work

The most common way to model the user in recommender systems is to use a vector of items ratings, and to provide recommendations comparing these vectors, in order to find similar users (collaborative-filtering approach) or similar items (content-based approach) [1]. Rather than using simple feature vector models and computing user similarity on this whole set of items, in our work the definition of user profiles is relative to the ontology, giving rise to *ontological user profiles*. This allows for association of user interests with relevant concepts in the ontology, rather than single atomic entities, and for a dynamic update of such user interests in the ontology.

Similar work exploiting ontological user profiles can be found in [16, 9].

As in our case, in [16] the ontological user profiles are instances of the pre-existing domain ontology. Similarly to us, their algorithm incrementally updates user profiles, based on the results of user interaction with the system. To update the interest values they exploit a spreading activation approach, thus treating the ontology as a semantic network, and propagate the values only in one direction. Instead, we perform a bi-directional propagation of the values. They do not take into consideration the amount of feedback on each node, the type of feedback received and the level of the node in the ontology.

Similar approach is the one of [9], where the user feedback on research papers is collected and effects the interest in ontological topics in the user model. Relationships between topics in the ontology are exploited to infer other topics of interest. Differently from us, they propagate the interest in a static manner: the interest value for a specific class is spread to the super-class always as 50% of its value. Another difference is that they use a time decay function, which weights more the recently seen papers than the older ones, while we weight more the past interaction with respect to the new one.

Another work which performs a similar value propagation in the ontology is the one of [15], where the authors use spreading activation techniques combined with classical search in order to improve the search process in a certain semantic domain. They apply their technique using a hybrid instances network, where each relation instance is assigned both a semantic label and a numerical weight based on certain properties of the ontology. The initial values for the spreading are obtained using classical search techniques on ontological concepts and used further to find related concepts. This technique makes it possible to find concepts which do not contain any of the words specified in the initial search. A point to notice is that they also use the constant that functions as attenuation factor, diminishing the activation with each processed node.

Other works propose the use of ontological user models in a different sense with respect to us. They create a specific ontology to represent the user features in the user model (demographic features such as age, gender, profession, etc). Examples are the General User Model Ontology (GUMO) [6] and the Unified User Context Model (UUCM) [8]. Differently from them we only use the ontology to model the domain, and represent the user model as an overlay over such a domain ontology. As a mixed approach which uses ontologies for user and domain modeling, we can cite Onto-bUM [12]. In addition to Domain Ontology, authors introduce the User Ontology which describes different features of the users, as well as the Log Ontology which describes the user interaction.

The propagation of values on an ontology can occur only among similar concepts. The similarity of two concepts in an ontology can be measured in different ways: by using the *information content* in an IS-A taxonomy (given by the negative logarithm of the probability of occurrence of the class in a text corpus) [14], by using the ontology graph structure, i.e. calculating the *distance between nodes* [11], by using RDF(S) or OWL DL primitives such as *rdf:id*, *owl:subClassOf* etc. to estimate partial similarities between concepts [2], etc. In this paper, we consider only the IS-A based similarity.

## 6 Conclusions and Future Work

This paper describes a promising approach to vertical propagation of user interests in an ontology-based user model. Starting from a given node in the ontology it is possible to propagate user interests to its ancestors and descendants. The novelty of our algorithm stems from the fact that we take into account the ontological structure of the domain, letting the level of each node influence the propagated interest values (the nodes lower down in the ontology can sense and propagate more). The high amount of past feedback for certain nodes helps decrease the propagation process for these nodes and prevents the non-proportional propagation of interests throughout the ontology. In addition, there is a possibility to treat various types of feedback differently, hence giving them different levels of importance.

The main contributions of our work are the following:

1. solution of the cold start problem - it is easier to obtain interest values for more domain items even with low amount of the user feedback;
2. alleviation of the sparsity problem - the number of items with associated interest values increases faster;
3. improvement of the recommendation accuracy - the items recommended to users mirror more closely their interests.

We tested our approach in a preliminary evaluation, obtaining satisfactory results for both downward and upward propagation. When producing the lists simulating user interests, 73% of the created lists for downward propagation and 61% for upward propagation had correlation coefficient $r \geq 0.5$. For the downward propagation, $r = 1$ for 18.5% of the cases and for the upward propagation $r \geq 0.9$ for 27% of the cases.

The next step will be the evaluation of our approach in more realistic conditions of usage of the system, in order to take into account other feedback typologies. We intend to test our framework at Bra "Cheese" festival in September 2011. We also plan on evaluating our approach in a different domain, represented by a different ontology, since the way the ontology is built can be the bottleneck of the approach. In fact, the approach is highly dependent on the knowledge representation.

Another possible future direction is to exploit ontological user profile and our propagation of interests to compute users' similarity, as in [17]. Since the propagated interest depends on the distance between the node that receives the direct feedback and the one that receives the propagated feedback, it would be interesting to see how our approach would behave when the notion of exponentially decreasing edge lengths in the ontology is used (see [5]). Another interesting feature that we would like to explore is adding constraints to our propagation algorithm along the lines of [15], such as concept type constraints (no propagation to certain kinds of nodes) or distance constraints (stopping the propagation upon reaching the nodes too far from the initial node). Note that in this version we are not considering compound concepts modeled by "part-whole" relations. We intend to achieve this by considering properties of domain items in the ontology, thus enabling also "horizontal propagation" among concepts. Finally, we plan on introducing some temporal aspects to the propagation mechanism.

## References


1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
2. T.-L. Bach and R. Dieng-Kuntz. Measuring Similarity of Elements in OWL DL Ontologies. In *Proc. of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.
3. P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996.
4. F. Carmagnola, F. Cena, L. Console, O. Cortassa, C. Gena, A. Goy, I. Torre, A. Toso, and F. Vernero. Tag-based user modeling for social multi-device adaptive guides. *User Modeling and User-Adapted Interaction*, 18:497–538, 2008.
5. E. Chiabrando, S. Likavec, I. Lombardi, C. Picardi, and D. Theseider Dupré. Semantic similarity in heterogeneous ontologies. In *Proc. of the 22nd ACM Conference on Hypertext and Hypermedia, Hypertext '11*, pages 153–160. ACM, 2011.
6. D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. GUMO - the General User Model Ontology. In *Proc. of the 10th Int. Conference on User Modeling, UM '05*, volume 3538 of *LNCS*, pages 428–432. Springer, 2005.
7. A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationship. *The Knowledge Engineering Review*, 16(2):111–155, 2001.
8. B. Mehta, C. Niederée, A. Stewart, M. Degemmis, P. Lops, and G. Semeraro. Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In *Proc. of the 10th Int. Conference on User Modeling, UM '05*, volume 3538 of *LNCS*, pages 119–123. Springer, 2005.
9. S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22:54–88, 2004.
10. PIEMONTE Team. WantEat: interacting with social networks of intelligent things and people in the world of enogastronomy. In *Interacting with Smart Objects Workshop 2011*, 2011.
11. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems Management and Cybernetics*, 19(1):17–30, 1989.
12. L. Razmerita, A. Angehrn, and A. Maedche. Ontology-based user modeling for knowledge management systems. In *Proc. of the 9th Int. Conference on User modeling, UM '03*, volume 2702 of *Lecture Notes in Computer Science*, pages 213–217. Springer, 2003.
13. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14th Int. Joint Conference on Artificial Intelligence*, pages 448–453, 1995.
14. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
15. C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proc. of the 13th Int. Conference on World Wide Web*, WWW '04, pages 374–383. ACM, 2004.
16. A. Sieg, B. Mobasher, and R. Burke. Web search personalization with ontological user profiles. In *Proc. of the 16th ACM Conference on Information and Knowledge Management*, CIKM '07, pages 525–534. ACM, 2007.
17. A. Sieg, B. Mobasher, and R. Burke. Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In *Proc. of the 1st Int. Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, pages 39–46. ACM, 2010.
18. S. Stevens. On the psychophysical law. *Psychological Review*, 64(3):153–181, 1957.